



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/057,002	01/24/2002	Shlomo Wygodny	MUTEK.4C1CP1	5496

20995 7590 08/25/2005

KNOBBE MARTENS OLSON & BEAR LLP  
2040 MAIN STREET  
FOURTEENTH FLOOR  
IRVINE, CA 92614

EXAMINER

MITCHELL, JASON D

ART UNIT	PAPER NUMBER
----------	--------------

2193

DATE MAILED: 08/25/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

10/057,002

Applicant(s)

WYGODNY ET AL.

Examiner

Jason Mitchell

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 01 June 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-36 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-36 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

1. This action is in response to remarks filed on 6/1/05.
2. Claims 1, 3, 8, 27 and 31 have been amended. Claims 1-36 are pending in this case.

### ***Response to Arguments***

3. **Applicant's arguments, on pg. 8 regarding the USC 112 1<sup>st</sup> paragraph rejection of claim 3 have been fully considered and are persuasive. The rejection has been withdrawn.**
4. **Applicant's arguments on pp. 9-10 regarding the Obviousness-Type Double Patenting Rejection of claims 6-35 have been fully considered but they are not persuasive.**

In the first full paragraph on pg. 10 Applicant states:

Thus, the system of Buzbee is directed to dynamically-translated code operating under the control of a debugger. There is no suggestion to combine the dynamic translation and debugging system of Buzbee with the system and method for remotely analyzing the execution of computer programs of Wygodny. Applicants respectfully assert the combination suggested by the examiner is based on hindsight.

Examiner respectfully disagrees. At the time of the invention one of ordinary skill in the art would have recognized that debugging is a means of analyzing the execution of computer programs, consequently, the teachings of Buzbee are found in an analogous art. Further, as noted in the rejection, Buzbee's 'conditional breakpoints' provide a means of reporting only information that a developer deems useful thereby easing and

Art Unit: 2193

speeding the process of debugging or analyzing the program and providing the motivation for the combination.

**Applicant's arguments on pg. 10 regarding the USC 102(b) rejection of claims 1, 2 and 5 have been fully considered but they are not persuasive.**

In the second paragraph under the heading 'Rejection of Claims 1-2 and 5 Under 35

USC 102(b), Applicant states:

In the system described by Blasciak, code markers are inserted into the source code. The source then must be recompiled (see e.g., Figure 4, items 402, 404, 406, and 408). By contrast, the method claimed by Applicants attaches to an executable module loaded in memory.

Examiner respectfully disagrees. Blasciak clearly discloses that his methods do not require recompilation as suggested by Applicant (col. 14, lines 45-49 'the code markers may be inserted without recompiling the entire program'). Blasciak further discloses a method of dynamically inserting code markers in executable module loaded in memory (col. 14, lines 45-49 'adding the ability to insert ... code markers dynamically').

Applicant goes on to restate the limitations of claims 1, 2 and 5, but fails to specifically point out how the language of the claims patentably distinguishes them from the references.

Accordingly, the USC 102(b) rejections of claims 1, 2 and 5 are maintained.

**Applicant's arguments on pg. 11 regarding the USC 103(a) rejection of claim 4 have been fully considered but they are not persuasive.**

Art Unit: 2193

The USC 103(a) rejection of claim 4 is maintained for the same reasons given in regard to the rejection of claims 1,2 and 5.

### ***Claim Rejections - 35 USC § 112***

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. **Claim 30 recites the limitation "said trace information" in line 10. There is insufficient antecedent basis for this limitation in the claims.** Examiner's best understanding will be used in the treatment of this claim and "said trace information" will be assumed to refer to "said trace data".

### ***Double Patenting***

7. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

**8. Claims 6-35 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-32 of U.S. Patent No. 6,202,199 (199) in view of US 5,732,210 to Buzbee (Buzbee).**

**Regarding Claim 6:** Claim 2 of the 199 reference discloses the limitations in the instant claim except the limitation of the trace control information including one or more pairs of triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 7:** Claim 1 of the 199 reference discloses the limitations in the instant claim except the limitation of the trace control information including one or more pairs of triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is

Art Unit: 2193

satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 8:** Claim 5 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 9:** Claim 6 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 10:** Claim 11 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').



Art Unit: 2193

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 11:** Claim 12 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 12:** Claim 7 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 13:** Claim 13 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

Art Unit: 2193

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 14:** Claim 14 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 15:** Claim 15 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 16** Claim 16 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

Art Unit: 2193

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 17:** Claim 18 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 18:** Claim 19 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace control information including specification of at least one trigger/action pair and said trace information being generated by specified actions in response to specified triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 19:** Claim 11 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace control information including specification of at least one trigger/action pair and said trace information being generated by specified actions in response to specified triggers.

Art Unit: 2193

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 20:** Claim 11 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace control information including specification of at least one trigger/action pair and said trace information being generated by specified actions in response to specified triggers and an occurrence of an event corresponding to execution of a source line.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action') and that said triggers are in response

Art Unit: 2193

to the execution of a source line (col. 2, lines 6-8 'when it reaches a certain code location').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 21:** Claim 11 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace control information including specification of at least one trigger/action pair and said trace information being generated by specified actions in response to specified triggers and that said action includes writing a comment to said trace log.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action') wherein said action includes writing a comment to said trace log (col. 2, lines 1-2 'the debugger is then instructed to print the value of "i"').



Art Unit: 2193

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 22:** Claim 11 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace control information including specification of at least one trigger/action pair and said trace information being generated by specified actions in response to specified triggers and said triggers including at least one condition, said condition specifying whether said trigger causes the execution of said at least one action.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action') and that said triggers include at least one condition, said condition specifying whether said trigger causes the execution of said at least one action (col. 2, lines 8-10 'If the condition is satisfied, the debugger ... performs some specified action').

Art Unit: 2193

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 23:** Claim 5 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers and said condition comprises a logical expression based on a variable.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action') said event comprises a logical expression based on a variable (col. 2, lines 14-16 'if "i" equals 3').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a

Art Unit: 2193

breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 24:** Claim 5 of the 199 reference discloses the limitations in the instant claim except the limitation of said trace information being generated by specified actions in response to specified triggers and said condition comprises a logical expression based on a variable and said trace log information is based on conditional trace control information.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action') said event comprises a logical expression based on a variable (col. 2, lines 14-16 'if "i" equals 3') and generating a log (col. 2, lines 1-2 'debugger is then instructed to print the value of "i"') based on conditional trace control information (col. 2, lines 8-10 'If the condition is satisfied').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied, the debugger halts execution').

**Regarding Claim 25:** Claim 22 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 26:** Claim 23 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

Art Unit: 2193

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 27:** Claim 26 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been

Art Unit: 2193

motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 28:** Claim 24 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 29:** Claim 25 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Art Unit: 2193

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 30:** Claim 28 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by

Art Unit: 2193

Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 31:** Claim 29 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').



Art Unit: 2193

**Regarding Claim 32:** Claim 30 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 33:** Claim 31 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is

Art Unit: 2193

satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 34:** Claim 32 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ...

Art Unit: 2193

performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 35:** Claim 33 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking actions in response to triggers and said trace control information comprising triggers and actions.

Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

**Regarding Claim 36:** Claim 33 of the 199 reference discloses the limitations in the instant claim except the limitation of wherein collecting trace data comprises taking

Art Unit: 2193

actions in response to triggers and said trace control information comprising triggers and actions and a trace tree showing the results of actions taken in response to triggers. Buzbee teaches pairs of triggers and actions (col. 2, lines 4-5 'conditional breakpoints'), said triggers and actions each specifying an event (col. 2, lines 8-10 'If the condition is satisfied') and an action to take in response to said event (col. 2, lines 8-10 'halts execution ... or performs some specified action').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include triggers (col. 2, lines 4-5 'conditional breakpoints') as taught by Buzbee in the 'trace control information' disclosed in the 199 reference and to use the conditional actions (col. 2, lines 8-10 'If the condition is satisfied the debugger ... performs some specified action') taught by Buzbee to generate the trace information disclosed in the 199 reference, because one of ordinary skill in the art would have been motivated to provide the developer with the flexibility of disregarding a breakpoint except under certain conditions (col. 2, lines 8-10 'If the condition is satisfied').

### ***Claim Rejections - 35 USC § 102***

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

10. **Claims 1-2 and 5 are rejected under 35 U.S.C. 102(b) as being anticipated by US 5,265,254 to Blasciak (Blasciak).**

Art Unit: 2193

**Regarding Claim 1:** Blasciak discloses a method of tracing the execution of a computer program comprising: generating trace control information which specifies triggers and associated actions (col. 7, lines 33-37 'the code marked by the software engineer'); attaching to a computer program loaded into a computer memory for execution (col. 2, lines 59-61 'inserted ... interactively during the debug session') tracing executing of the computer program according to the trace control information, such that when one of said triggers occurs the corresponding action is performed (col. 3, lines 29-34 'write on procedure entry and exit'); and generating a trace log of said tracing, wherein the trace log reflects said actions performed during tracing (col. 8, lines 28-34 'a time-stamped trace list').

**Regarding Claim 2:** The rejection of claim 1 is incorporated; further Blasciak discloses said triggers include at least the execution of the assembly code generated from a function entry or exit (col. 3, lines 29-34 'write on procedure entry and exit').

**Regarding Claim 5:** the rejection of claim 1 is incorporated; further Blasciak discloses said actions include at least writing to said trace log the stack dump of functions active at a time of a trigger (col. 3, lines 29-34 'task table dumps').

### ***Claim Rejections - 35 USC § 103***

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

**12. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over US 5,265,254 to Blasciak (Blasciak) in view of 6,321,375 Blandy (Blandy).**

**Regarding Claim 3:** The rejection of claim 1 is incorporated; further Blasciak does not explicitly disclose that a trigger comprises expiration of a timer.

Blandy teaches a trigger which comprises expiration of a timer (col. 34, lines 45-46 'a timer based sampling process') in an analogous art for the purpose of monitoring execution of a program (col. 34, 45-46 'to track executing methods').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use timer based triggers as taught in Blandy (col. 34, lines 45-46) in conjunction with the code markers taught in Blasciak (col. 2, lines 57-59 'code markers') thereby providing additional capabilities to the execution monitoring system (Blasciak Abstract 'observing indications ... of the occurrence of an event in the execution of the software').

**13. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over US 5,265,254 to Blasciak (Blasciak) in view of 5,732,210 to Buzbee (Buzbee).**

**Regarding Claim 4:** The rejection of claim 1 is incorporated; further Blasciak discloses at least some of said actions are activated only if a plurality of conditions are satisfied while said triggers are activated (col. 8, lines 36-38 'Other parameter contents may also be monitored'), but does not disclose that said conditions including comparing values of data passed in said execution to other values, however Blasciak does disclose the use of conventional breakpoint technology (col. 8, lines 36-41 'such as break points ...

Art Unit: 2193

Implementation of such measurements are believed to be well within the skill of those skilled in the art').

Buzbee teaches that triggers ('conditional breakpoints') may include comparing values of data passed in said execution to other values (col. 2, lines 14-16 'if "i" equals 3') in an analogous art for the purpose of providing a debugger the ability to ignore triggers if the state of execution is not of interest (col. 2, lines 10-11 'If the condition is not satisfied, the debugger allows the program to continue').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use triggers ('conditional breakpoints') as taught by Buzbee to provide the code markers of Blasciak the ability to ignore triggers if the state of execution is not of interest (col. 2, lines 10-11 'If the condition is not satisfied, the debugger allows the program to continue')

### ***Conclusion***

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. 6,374,369 to O'Donnell discloses timer-based breakpoints.

15. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not

Art Unit: 2193

mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Mitchell whose telephone number is (571) 272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Jason Mitchell  
8/9/05



KAKALI CHAKI  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100